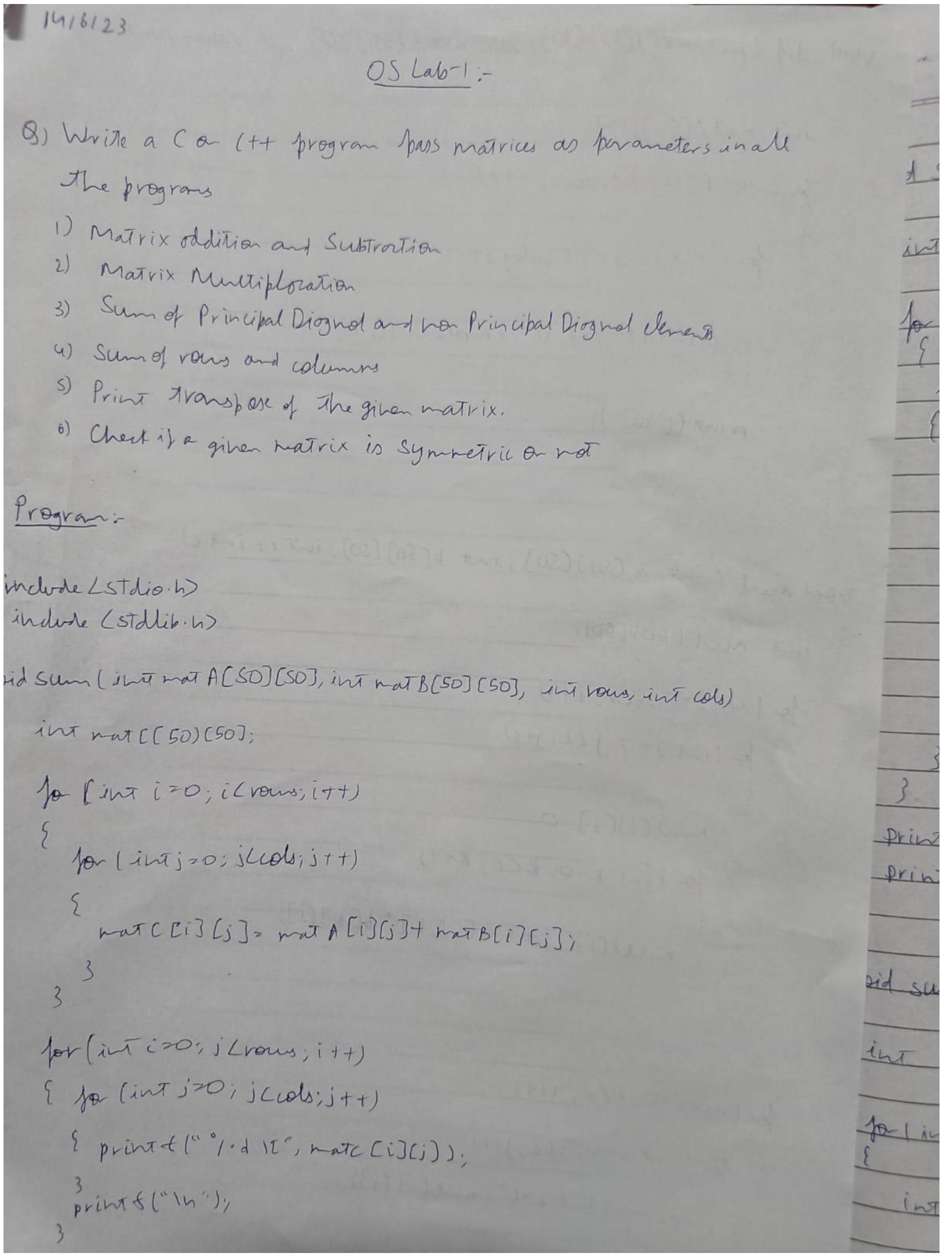
OS LAB 1:



14/6/23

OS Lab-1 :-

Q) Write a C or C++ program pass matrices as parameters in all the programs

1) Matrix addition and Subtraction
2) Matrix Multiplication
3) Sum of Principal Diognol and non Principal Diognol elements
4) Sum of rows and columns
5) Print transpose of the given matrix.
6) Check if a given matrix is symmetric or not

Program :-

```
include <STdio.h>
 include <stdlib.h>

id sum ( int mat A[50][50], int mat B[50][50],  int rows, int cols)
   int mat C[50][50];

   for (int i=0; i<rows; i++)
   {
      for (int j=0; j<cols; j++)
      {
         mat C [i][j] = mat A [i][j] + mat B[i][j];
      }
   }

   for (int i=0; i<rows; i++)
   { for (int j=0; j<cols; j++)
      { printf(" %d \t", matC [i][j]);
      }
      printf("\n");
   }
```

```
void dif (int mat A[50][50], int mat B[50][50], int rows, int cols)
{
    int mat C[50][50];
    for ( int i=0 ; j<rows; i++)
    {
       for ( int j=0; j<cols; j++)
       {
           printf("%d \t" , matC [i][j]);
       }
       printf("\n");
     }
 }

void mul ( int a [50][50] , int b[50][50], int r, int c)
{ int mul [50][50];
   for (int i=0; i<r; i++)
   { for (int j=0; j<c; j++)
      {
          mul[i][j]=0;
          for (int k=0; k<c; k++)
          {
            mul[i][j] += a[i][k] * b[k][j];
           }
       }
    }
   for (int i=0; i<r; i++)
   { for ( int j=0; j<c; j++)
      {   printf("%d\t", mul[i][j]);
      }
     printf("\n");
```

```
void sumd (int matA[50][50], int rows, int cols)
{
  int sumd=0, sumnd=0;

  for (int i=0; i<rows; i++)
   {
    for ((int j=0; j<cols; j++)
    {
      if(i==j)
       {
         sumd = sumd+ matA[i][j];
       }
      if(i+j == rows -1)
      {
        sumnd= sumnd + matA[i][i];
      }
    }
  }
  print("SUM of Diagnol : %d\n", sumd);
  printf("SUM of Non principal diagnol : %d\n", sumnd);
}

void sumrc (int matA[50][50], int rows, int cols)
{
 int matC[50][50];

 for ( int i=0; i<rows; i++)
 {
   int j=0, sum=0;
for (j=0; j<cols; j++)
{
  Sum = Sum+ matA[j][i];
}
matC[j][i] =Sum;
}

for(int i=0; i<rows+1; i++)
{
  for (int j=0; j<cols+1; j++)
  {
    printf("%d\t", matC[i][j]);
  }
  printf("\n");
}
}

void transp (int matA[50][50], int rows, int cols)
{
  for(int i=0; i<rows; i++)
  {
    for(int j=0; j<cols; j++)
    {
      printf("%d\t", matA[j][i]);
    }
    printf("\n");
  }
}
```

Date :
Page No :

```
void symm (int matA [50][50], int rows , int cols)
(
 int flag = 0;
  for( int i=0; i<rows; i++)
  {
    for (int j=0; j<cols; j++)
    {
     if (mat A [i][j] == mat A[j][i])
     {
       continue;
     }
     else
     {
        printf(" Not a Transpose matrix");
        return;
     }
    }
  }
 printf(" Transpose matrix");
}


int main()
{
  int opt;
  Scanf("%d", &opt);

  while(1)
  {
    print("1. Add and Sub 2. Multiply 3. Sum of principal and
non principal diagnol elements 4. Sum of rows and columns.
5. Transpose of Matrix 6. Matrix Symmetrical");
```

```
printf("Enter option:");
scanf("%d", &opt);

Switch(opT)
{
 Case 0: case 1:
          Sum(mat A[50], mat B, 3,3);
          break diff(matA, mat B,3,3);
          break;

Case 2:
     mull(mat A, mat B, 3,3);
      break;

case 3:
     Sumd(mat A, 3,3);
     break;

case 4:
    Sumrc(matA, 3,3);
    break;

Case 5:
    transp(matA, 3,3);
    break;

Case 6:
    Symm(mat A, mat B, 3,3);
    break;

Case 7: exit(0); }
```

```
}

return 0;
}
```

O/P:-

```
Matrix A:- 0 1 2
           3 4 5
           6 7 8

Matrix B:- 1 2 3
           3 4 5
           5 6 7

Sum:-  1  3  5
       6  8 10
      11 13 15

Difference:- -1 -1 -1
              0  0  0
              1  1  1

Multiplcation:- 13 16  19
                40 52  64
                67 70 109

Sum of Diagnols:12
Sum of Non principal diagnols:12
```

Sum of rows and columns:-

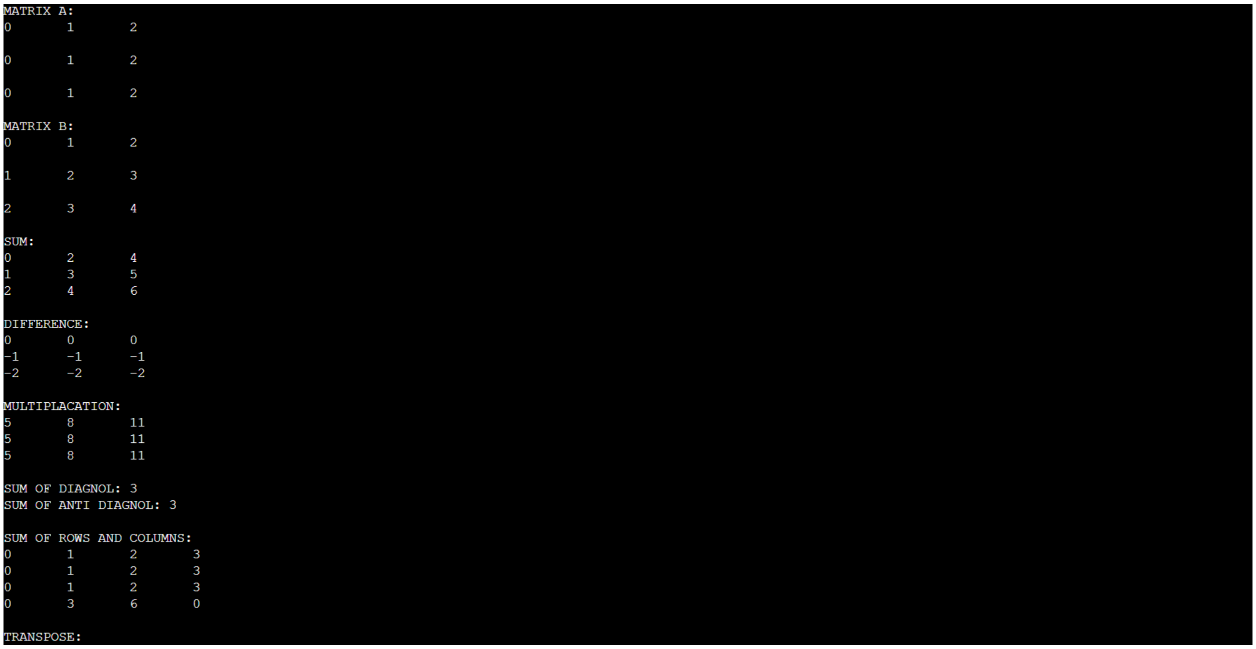
$$\begin{matrix} 0 & 1 & 2 & 3 \\ 3 & 4 & 5 & 12 \\ 6 & 7 & 8 & 21 \\ 9 & 12 & 15 & 0 \end{matrix}$$

Transpose:-

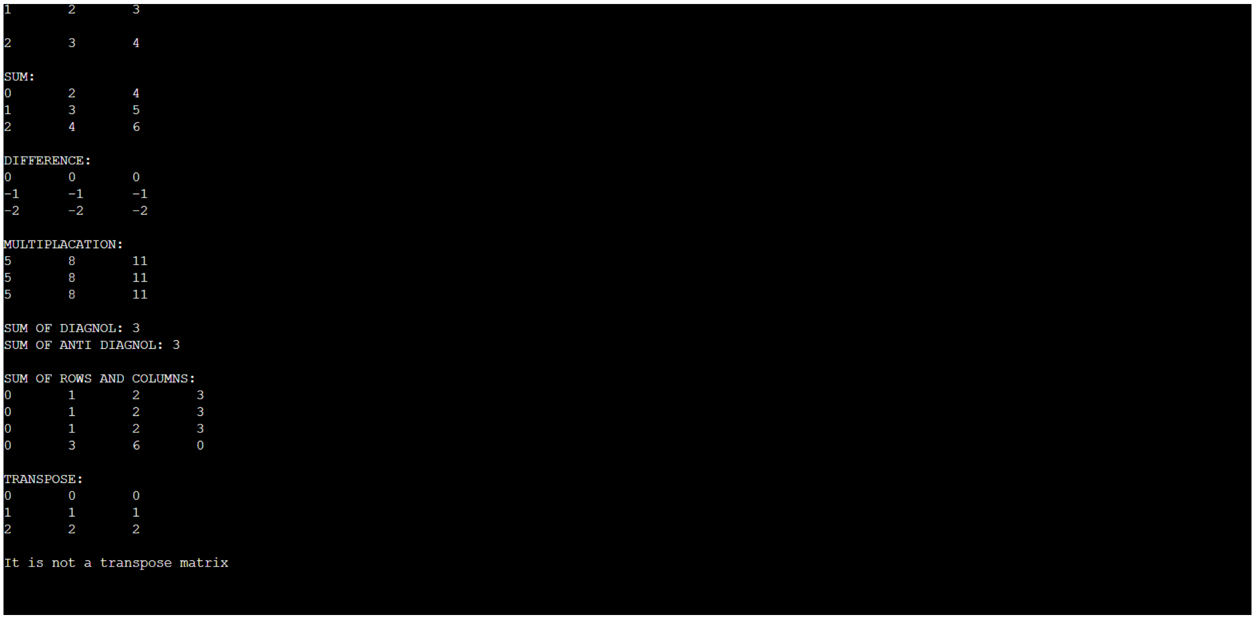
$$\begin{matrix} 0 & 3 & 6 \\ 1 & 4 & 7 \\ 2 & 5 & 8 \end{matrix}$$

Matrix A is not symmetric.

Output:



```
MATRIX A:
0       1       2

0       1       2

0       1       2

MATRIX B:
0       1       2

1       2       3

2       3       4

SUM:
0       2       4
1       3       5
2       4       6

DIFFERENCE:
0       0       0
-1      -1      -1
-2      -2      -2

MULTIPLACATION:
5       8       11
5       8       11
5       8       11

SUM OF DIAGNOL: 3
SUM OF ANTI DIAGNOL: 3

SUM OF ROWS AND COLUMNS:
0       1       2       3
0       1       2       3
0       1       2       3
0       3       6       0

TRANSPOSE:
```



```
1       2       3

2       3       4

SUM:
0       2       4
1       3       5
2       4       6

DIFFERENCE:
0       0       0
-1      -1      -1
-2      -2      -2

MULTIPLACATION:
5       8       11
5       8       11
5       8       11

SUM OF DIAGNOL: 3
SUM OF ANTI DIAGNOL: 3

SUM OF ROWS AND COLUMNS:
0       1       2       3
0       1       2       3
0       1       2       3
0       3       6       0

TRANSPOSE:
0       0       0
1       1       1
2       2       2

It is not a transpose matrix
```